EXHIBIT H

JENAM TECH LLC'S INFRINGEMENT ANALYSIS

U.S. Patent No. 10,069,945 – Google LLC
Claim 104

Jenam Tech LLC ("Jenam") provides evidence of infringement of claim 104 of U.S. Patent No. 10,069,945 (hereinafter "the '945 patent") by Google LLC ("Google"). In support thereof, Jenam provides the following claim charts.

"Accused Instrumentalities" as used herein refers to at least one or more websites or web addresses including, but not limited to www.google.com, stored and/or hosted on one or more servers owned or under the control of Google. These claim charts demonstrate Google's infringement, and provide notice of such infringement, by comparing each element of the asserted claims to corresponding components, aspects, and/or features of the Accused Instrumentalities. These claim charts are not intended to constitute an expert report on infringement. These claim charts include information provided by way of example, and not by way of limitation.

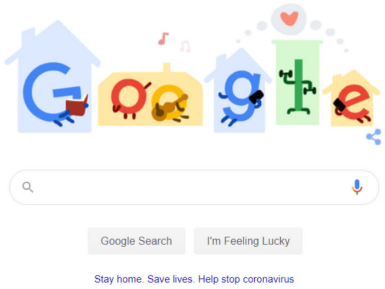
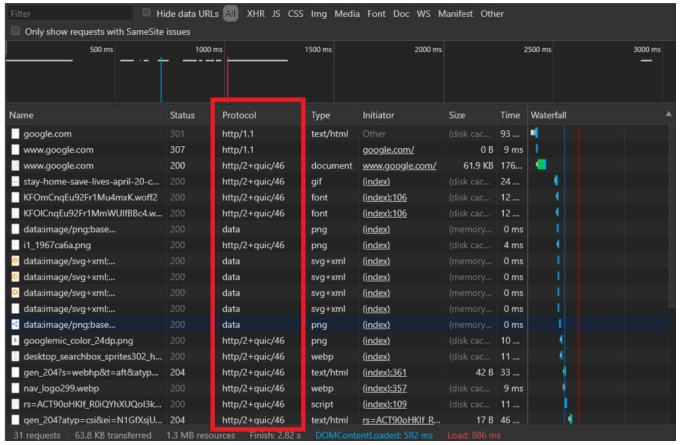
The analysis set forth below is based only upon information from publicly available resources regarding the Infringing Instrumentalities, as Google has not yet provided any non-public information. An analysis of Google's (or other third parties') technical documentation and/or software source code may assist in fully identify all infringing features and functionality. Accordingly, Jenam reserves the right to supplement this infringement analysis once such information is made available to Jenam. Furthermore, Jenam reserves the right to revise this infringement analysis, as appropriate, upon issuance of a court order construing any terms recited in the asserted claims.

Unless otherwise noted, Jenam contends that Google directly infringes the '945 patent in violation of 35 U.S.C. § 271(a) by selling, offering to sell, making, using, and/or importing the Infringing Instrumentalities. The following exemplary analysis demonstrates that infringement. Unless otherwise noted, Jenam further contends that the evidence below supports a finding of indirect infringement under 35 U.S.C. §§ 271(b) and/or (c), in conjunction with other evidence of liability under one or more of those subsections. Google makes, uses, sells, imports, or offers for sale in the United States, or has made, used, sold, imported, or offered for sale in the past, without authority, or induces others to make, use, sell, import, or offer for sale in the United States, or has induced others to make, use, sell, import, or offer for sale in the past, without authority products, equipment, or services that infringe claim 104 of the '945 patent, including without limitation, the Accused Instrumentalities.

Unless otherwise noted, Jenam believes and contends that each element of each claim asserted herein is literally met through Google's provision of the Infringing Instrumentalities. However, to the extent that Google attempts to allege that any asserted claim element is not literally met, Jenam believes and contends that such elements are met under the doctrine of equivalents. More specifically, in its investigation and analysis of the Infringing Instrumentalities, Jenam did not identify any substantial differences between the elements of the patent claims and the corresponding features of the Infringing Instrumentalities, as set forth herein. In each instance, the identified feature of the Infringing Instrumentalities performs at least substantially the same function in substantially the same way to achieve substantially the same result as the corresponding claim element.

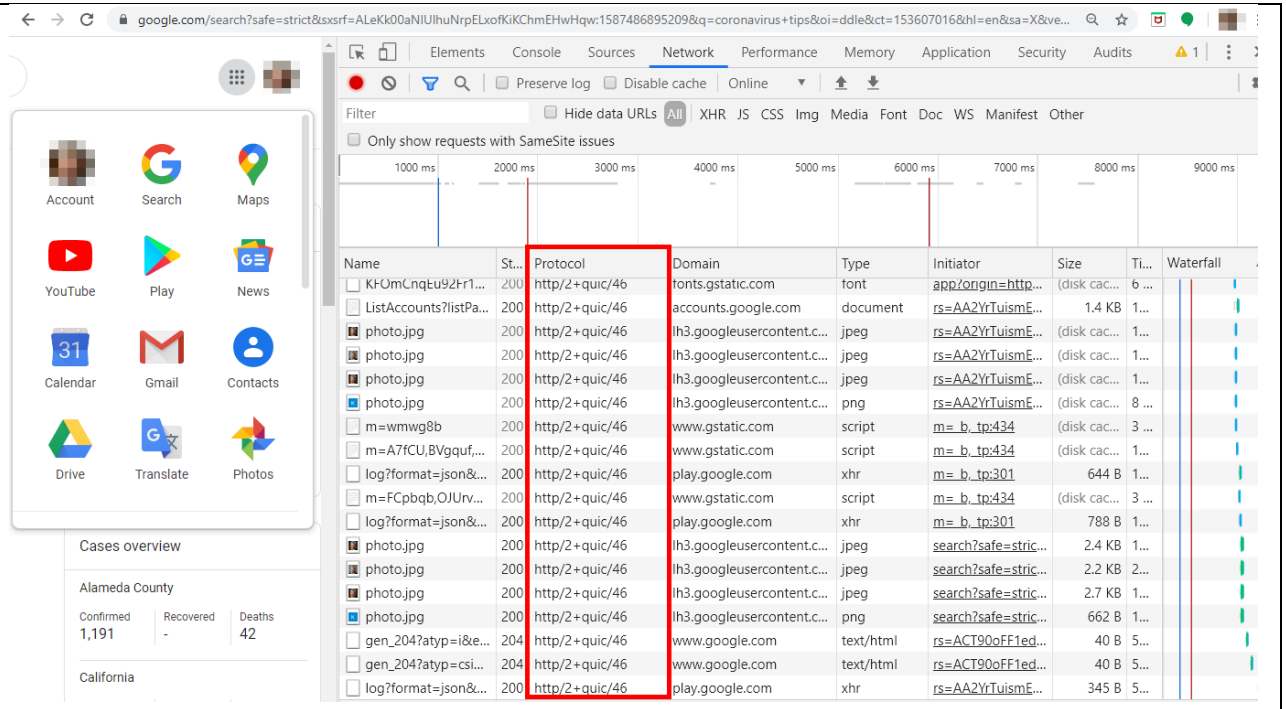
To the extent the chart of an asserted claim relies on evidence about certain specifically-identified Accused Instrumentalities, Jenam asserts that, on information and belief, any similarly-functioning instrumentalities also infringes the charted claim. Jenam reserves the right to amend this infringement analysis based on other products made, used, sold, imported, or offered for sale by Google. Jenam also reserves the right to amend this infringement analysis by citing other claims of the '945 patent, not listed in the claim chart, that are infringed by the Accused Instrumentalities. Jenam further reserves the right to amend this infringement analysis by adding, subtracting, or otherwise modifying content in the "Accused Instrumentalities" column of each chart.

CLAIM CHARTS
BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
U.S. Patent No. 10,069,945

Claim 104	Accused Instrumentalities
A computer-implemented method comprising: providing access to a server computer including: a non-transitory memory storing instructions, and one or more processors in communication with the non-transitory memory, wherein the one or more processors execute the instructions such that a network application operates in accordance with a first protocol including a transmission control protocol (TCP) that operates above an Internet Protocol (IP) layer and below a hypertext transfer protocol (HTTP) application layer, the server computer configured to operate in accordance with the first protocol to set up a TCP connection with a client computer;	Google owns or controls a server computer that performs a method including providing access to the server computer that includes: a non-transitory memory storing instructions and one or more processors in communication with the non-transitory memory. The one or more processors execute the instructions such that a network application (e.g., server software, etc.) operates in accordance with a first protocol including a transmission control protocol (TCP) that operates above an Internet Protocol (IP) layer and below a hypertext transfer protocol (HTTP) application layer. The server computer is configured to operate in accordance with the first protocol to set up a TCP connection with a client computer. See excerpt(s) below, for example (emphasis added, if any): **Note**: Below is a web page of Google (https://www.google.com/).  
causing first data to be communicated from the server computer to the client computer utilizing the TCP connection in accordance with the TCP protocol and a hypertext transfer protocol	Google owns or controls the server computer that performs the method including causing first data to be communicated from the server computer to the client computer utilizing the TCP connection in accordance with the TCP protocol and a hypertext transfer protocol (HTTP), for being presented to a user of the client computer. See excerpt(s) below, for example (emphasis added, if any):

(HTTP), for being presented to a user of the client computer;	**Note**: The aforementioned web page of Google (https://www.google.com/) is configured for causing first data to be communicated from the server computer to the client computer utilizing the TCP connection in accordance with the TCP protocol and a hypertext transfer protocol (HTTP), for being presented to a user of the client computer.
causing the server computer to permit second data, of the user of the client computer, to be received at the server computer from the client computer utilizing the TCP connection in accordance with the TCP protocol and the hypertext transfer protocol (HTTP); and	Google owns or controls the server computer that performs the method including causing the server computer to permit second data, of the user of the client computer, to be received at the server computer from the client computer utilizing the TCP connection in accordance with the TCP protocol and the hypertext transfer protocol (HTTP). See excerpt(s) below, for example (emphasis added, if any): **Note**: The aforementioned web page of Google (https://www.google.com/) is configured for causing the server computer to permit second data, of the user of the client computer, to be received at the server computer from the client computer utilizing the TCP connection in accordance with the TCP protocol and the hypertext transfer protocol (HTTP).
providing access to structured data that results in the client computer operating in accordance with a second protocol, that is different from the TCP and operates above the IP layer and below the hypertext transfer protocol (HTTP) application layer, in order to setup a second protocol connection with another server computer, and to:	Google owns or controls the server computer that performs the method including providing access to structured data (e.g., in HTML pages, etc.) that results in the client computer operating in accordance with a second protocol (e.g., QUIC protocol, etc.), that is different from the TCP and operates above the IP layer and below the hypertext transfer protocol (HTTP) application layer, in order to setup a second protocol connection with another server computer. See excerpt(s) below, for example (emphasis added, if any): **Note**: Below is a web page of Google (https://www.google.com/).

CLAIM CHARTS
BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
U.S. Patent No. 10,069,945



Note: As set forth below, QUIC is different from TCP.

1. Introduction

QUIC is a multiplexed and secure transport protocol that runs on top of UDP. QUIC aims to provide a flexible set of features that allow it to be a general-purpose transport for multiple applications.

QUIC implements techniques learned from experience with TCP, SCTP and

"On the surface, QUIC is very similar to TCP+TLS+HTTP/2 implemented on UDP. ...However, since QUIC is built on top of UDP, it suffers from no such limitations." https://www.chromium.org/quic

Note: As set forth below, a QUIC negotiation packet is received by the client node from a server node.

	7.4. Transport Parameters During connection establishment, both endpoints make authenticated declarations of their transport parameters. These declarations are made unilaterally by each endpoint. Endpoints are required to comply with the restrictions implied by these parameters; the description of each parameter includes rules for its handling. QUIC packet: A well-formed UDP payload that can be parsed by a QUIC receiver. QUIC packet size in this document refers to the UDP **Note**: As set forth below, prior to a QUIC connection being established, the QUIC connection is "set up" using the aforementioned handshake. 3.1. Low-Latency Connection Establishment QUIC relies on a combined cryptographic and transport handshake for setting up a secure transport connection. QUIC connections are expected to commonly use 0-RTT handshakes, meaning that for most QUIC connections, data can be sent immediately following the client handshake packet, without waiting for a reply from the server. QUIC provides a dedicated stream (Stream ID 0) to be used for performing the cryptographic handshake and QUIC options negotiation. The format of the QUIC options and parameters used during negotiation are described in this document, but the handshake protocol that runs on Stream ID 0 is described in the accompanying cryptographic handshake draft [QUIC-TLS].
receive idle information for use in detecting an idle time period during which no signal is communicated that meets each of the following criteria: a) communicated in the second protocol connection, and b)	Google owns or controls the server computer that performs the method including providing access to the structured data (e.g., in the HTML pages, etc.) that results in the client computer operating in accordance with the second protocol (e.g., the QUIC protocol, etc.), in order to: receive idle information for use in detecting an idle time period (e.g., idle timeout parameter field, etc.) during which no signal is communicated that meets each of the following criteria: a) communicated in the second protocol connection, and b) results in the second protocol connection being at least partially kept alive (e.g., the connection remains open, etc.).

results in the second protocol connection being at least partially kept alive,	**Note**: The idle-timeout period is found in a QUIC negotiation packet. See excerpt(s) below, for example (emphasis added, if any): **Note**: As set forth below, a QUIC negotiation packet includes transport parameters that include an idle timeout parameter that is detected by a recipient of such packet. 7.4. Transport Parameters During connection establishment, both endpoints make authenticated declarations of their transport parameters. These declarations are made unilaterally by each endpoint. Endpoints are required to comply with the restrictions implied by these parameters; the description of each parameter includes rules for its handling.

Value	Parameter Name	Specification
0x0000	initial_max_stream_data	Section 7.4.1
0x0001	initial_max_data	Section 7.4.1
0x0002	initial_max_stream_id	Section 7.4.1
0x0003	idle_timeout	Section 7.4.1
0x0004	omit_connection_id	Section 7.4.1
0x0005	max_packet_size	Section 7.4.1
0x0006	stateless_reset_token	Section 7.4.1

Table 4: Initial QUIC Transport Parameters Entries

	The format of the transport parameters is the TransportParameters struct from Figure 6. This is described using the presentation language from Section 3 of [I-D.ietf-tls-tls13]. uint32 QuicVersion; enum { initial_max_stream_data(0), initial_max_data(1), initial_max_stream_id(2), idle_timeout(3), omit_connection_id(4), max_packet_size(5), stateless_reset_token(6), (65535) } TransportParameterId; **Note**: As set forth below, since the idle_timeout value sets the duration of idleness, after which the connection is shutdown, a timeout attribute of the connection is necessarily modified based on the value received in the idle_timeout field of the connection negotiation packet. idle_timeout (0x0003): The idle timeout is a value in seconds that is encoded as an unsigned 16-bit integer. The maximum value is 600 seconds (10 minutes). 7.8. Connection Termination Connections should remain open until they become idle for a pre-negotiated period of time. A QUIC connection, once established, can be terminated in one of three ways:

	7.8.2. Idle Timeout A connection that remains idle for longer than the idle timeout (see Section 7.4.1) becomes closed. Either peer removes connection state if they have neither sent nor received a packet for this time. The time at which an idle timeout takes effect won't be perfectly synchronized on peers. A connection enters the draining period when the idle timeout expires. During this time, an endpoint that receives new packets MAY choose to restore the connection. Alternatively, an endpoint that receives packets MAY signal the timeout using an immediate close.
generate, based on the idle information, a second protocol packet including an idle time period parameter field identifying metadata that is specified in a number of seconds or minutes, and	Google owns or controls the server computer that performs the method including providing access to the structured data (e.g., in the HTML pages, etc.) that results in the client computer operating in accordance with the second protocol (e.g., the QUIC protocol, etc.), in order to: generate, based on the idle information, a second protocol packet including an idle time period parameter field (e.g., idle timeout parameter field, etc.) identifying metadata (e.g., the value of the idle timeout parameter field, etc.) that is specified in a number of seconds or minutes. See excerpt(s) below, for example (emphasis added, if any): **Note**: As set forth below, since the idle_timeout value sets the duration of idleness, after which the connection is shutdown, a timeout attribute of the connection is necessarily modified based on the value received in the idle_timeout field of the connection negotiation packet. idle_timeout (0x0003): The idle timeout is a value in seconds that is encoded as an unsigned 16-bit integer. The maximum value is 600 seconds (10 minutes).

	7.8. Connection Termination Connections should remain open until they become idle for a pre-negotiated period of time. A QUIC connection, once established, can be terminated in one of three ways: 7.8.2. Idle Timeout A connection that remains idle for longer than the idle timeout (see Section 7.4.1) becomes closed. Either peer removes connection state if they have neither sent nor received a packet for this time. The time at which an idle timeout takes effect won't be perfectly synchronized on peers. A connection enters the draining period when the idle timeout expires. During this time, an endpoint that receives new packets MAY choose to restore the connection. Alternatively, an endpoint that receives packets MAY signal the timeout using an immediate close.
send, from the client computer to another server computer and during the set up of the second protocol connection, the second protocol packet to provide the metadata to the another server computer, for use by the another server computer in determining a timeout attribute associated with the second protocol connection.	Google owns or controls the server computer that performs the method including providing access to the structured data (e.g., in the HTML pages, etc.) that results in the client computer operating in accordance with the second protocol (e.g., the QUIC protocol, etc.), in order to: send, from the client computer to another server computer and during the set up of the second protocol connection, the second protocol packet to provide the metadata (e.g., the value of the idle timeout parameter field, etc.) to the another server computer, for use by the another server computer in determining a timeout attribute associated with the second protocol connection. See excerpt(s) below, for example (emphasis added, if any): **Note**: As set forth below, a QUIC negotiation packet includes transport parameters that include an idle timeout parameter that is detected by a recipient of such packet.

7.4. Transport Parameters

During connection establishment, both endpoints make authenticated declarations of their transport parameters. These declarations are made unilaterally by each endpoint. Endpoints are required to comply with the restrictions implied by these parameters; the description of each parameter includes rules for its handling.

Value	Parameter Name	Specification
0x0000	initial_max_stream_data	Section 7.4.1
0x0001	initial_max_data	Section 7.4.1
0x0002	initial_max_stream_id	Section 7.4.1
0x0003	idle_timeout	Section 7.4.1
0x0004	omit_connection_id	Section 7.4.1
0x0005	max_packet_size	Section 7.4.1
0x0006	stateless_reset_token	Section 7.4.1

Table 4: Initial QUIC Transport Parameters Entries

The format of the transport parameters is the TransportParameters struct from Figure 6. This is described using the presentation language from Section 3 of [I-D.ietf-tls-tls13].

```
uint32 QuicVersion;

enum {
   initial_max_stream_data(0),
   initial_max_data(1),
   initial_max_stream_id(2),
   idle_timeout(3),
   omit_connection_id(4),
   max_packet_size(5),
   stateless_reset_token(6),
   (65535)
} TransportParameterId;
```

Note: As set forth below, since the idle_timeout value sets the duration of idleness, after which the connection is shutdown, a timeout attribute of the connection is necessarily modified based on the value received in the idle_timeout field of the connection negotiation packet.

```
idle_timeout (0x0003): The idle timeout is a value in seconds that
   is encoded as an unsigned 16-bit integer. The maximum value is
   600 seconds (10 minutes).
```

7.8. Connection Termination

Connections should remain open until they become idle for a pre-negotiated period of time. A QUIC connection, once established, can be terminated in one of three ways:

	7.8.2. Idle Timeout A connection that remains idle for longer than the idle timeout (see Section 7.4.1) becomes closed. Either peer removes connection state if they have neither sent nor received a packet for this time. The time at which an idle timeout takes effect won't be perfectly synchronized on peers. A connection enters the draining period when the idle timeout expires. During this time, an endpoint that receives new packets MAY choose to restore the connection. Alternatively, an endpoint that receives packets MAY signal the timeout using an immediate close.

Caveat: The notes and/or cited excerpts utilized herein are set forth for illustrative purposes only and are not meant to be limiting in any manner. For example, the notes and/or cited excerpts, may or may not be supplemented or substituted with different excerpt(s) of the relevant reference(s), as appropriate. Further, to the extent any error(s) and/or omission(s) exist herein, all rights are reserved to correct the same.